HP aC++/HP ANSI C バージョン A.06.12 リリースノート

第 7 版

HP Itanium® ベース システム

Manufacturing Part Number : 5991-7409

2007 年 2 月

Printed in U.S.A.

© Copyright 2007 Hewlett-Packard Development Company L.P.

ご注意

1. 本書に記載した内容は、予告なしに変更することがあります。
2. 本書は内容について細心の注意をもって作成いたしましたが、万一ご不審な点や誤り、記載もれなど、お気付きの点がございましたら当社までお知らせください。
3. 当社は、お客様の誤った操作に起因する損害については、責任を負いかねますのでご了承ください。
4. 当社では、本書に関して特殊目的に対する適合性、市場性などについては、一切の保証をいたしかねます。また、備品、性能などに関連した損傷についても保証いたしかねます。
5. 当社提供外のソフトウェアの使用や信頼性についての責任は負いかねます。
6. 本書の内容の一部または全部を、無断でコピーしたり、他のプログラム言語に翻訳することは法律で禁止されています。
7. 本製品パッケージとして提供した本書や媒体は本製品用だけにお使いください。プログラムをコピーする場合はバックアップ用だけにしてください。プログラムをそのままの形で、あるいは変更を加えて第三者に販売することは固く禁じられています。

U.S. Government License

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

著作権

©Copyright 2007 Hewlett-Packard Development Company, L.P.

本書には著作権によって保護されている内容が含まれています。本書の内容の一部または全部を著作者の許諾なしに複製、改変、および翻訳することは、著作権法下での許可事項を除き、禁止されています。

商標

Intel® および Itanium® は、米国ならびに他の国における Intel Corporation またはその関連会社の商標または登録商標です。

UNIX® は、The Open Group の登録商標です。

納入後の保証について

- 保証の期間は、ご購入時に当社よりお出しした見積書に記載された期間とします。保証サービスは、当社の定める休日を除く月曜日から金曜日までの、午前 8 時 45 分から午後 5 時 30 分の範囲で無料で行います。当社で定めたシステム製品については出張修理を行い、その他の製品については当社にご返却いただいた上での引取り修理となります。当社が定める地域以外における出張修理対象製品の修理は、保証期間中においても技術者派遣費が有料となります。
- ソフトウェア製品の保証は上記にかかわらず、下記に定める範囲とさせていただきます。
  - ソフトウェア製品およびマニュアルは当社が供給した媒体物の破損、資料の落丁およびプログラムインストラクションが実行できない場合のみ保証いたします。
  - バグおよび前記以外の問題の解決は、別に締結するソフトウェアサポート契約に基づいて実施されます。
- 次のような場合には、保証期間内でも修理が有料となります。
  - 取扱説明書等に記載されている保証対象外部品の故障の場合。
  - 当社が供給していないソフトウェア、ハードウェア、または補用品の使用による故障の場合。
  - お客様の不適当または不十分な保守による故障の場合。
  - 当社が認めていない改造、酷使、誤使用または誤操作による故障の場合。
  - 納入後の移設が不適切であったための故障または損傷の場合。
  - 指定外の電源 (電圧、周波数) 使用または電源の異常による故障の場合。
  - 当社が定めた設置場所基準に適合しない場所での使用、および設置場所の不適当な保守による故障の場合。
  - 火災、地震、風水害、落雷、騒動、暴動、戦争行為、放射能汚染、およびその他天災地変等の不可抗力的事故による故障の場合。
- 当社で取り扱う製品は、ご需要先の特定目的に関する整合性の保証はいたしかねます。また、そこから生じる直接的、間接的損害に対しても責任を負いかねます。
- 当社で取り扱う製品を組み込みあるいは転売される場合は、最終需要先における直接的、間接的損害に対しては責任を負いかねます。
- 製品の保守、修理用部品の供給期間は、その製品の製造中止後 5 年間とさせていただきます。

本製品の修理については取扱説明書に記載されている最寄の事業所へお問い合わせください。

出版履歴

出版の日付は、最新版ができるたびに変更します。内容の小さな変更に対しては、増刷の際に対応し、出版日の変更は行いません。マニュアルのパーツ番号は、改訂が行われるたびに変更します。

新版の作成は、記載内容の訂正もしくはドキュメント製品の変更にともなって行われます。お手元のマニュアルが最新のものか否かは、当社の営業担当に確認してください。

- 2003 年 6 月 第 1 版、5012-7849 (英語版 : 5187-2289)
- 2004 年 3 月 第 2 版、5990-6779 (英語版 : 5990-6777)
- 2004 年 9 月 第 3 版、5990-8188 (英語版 : 5990-8187)
- 2004 年 12 月 第 4 版、5991-0700 (英語版 : 5991-0633)
- 2005 年 9 月 第 5 版、5991-1082 (英語版 : 5991-1072)
- 2006 年 5 月 第 6 版、5991-4890 (英語版 : 5991-4889)
- 2007 年 2 月 第 7 版、5991-7409 (英語版 : 5991-7408)

原典

本書は『*HP aC++/HP ANSI C Release Notes Version A.06.12*』(HP Part No. 5991-7408) を翻訳したものです。

目次

1. HP aC++/HP ANSI C リリースノート

2. このバージョンの新機能

バージョン A.06.12 の新機能 9
オプション +Ofast (-fast) および +Ofaster 9
+Oinit_check および +check=uninit 間の相互作用 9
バージョン A.06.10 の新機能 10
HP Code Advisor 10
+cond_rodata オプション (廃止) 11
+[no]dep_name オプション (新規) 11
+expand_types_in_diag オプション (新規) 11
+FPmode オプション (拡張) 11
+Ointeger_overflow オプション (デフォルト値の変更) 11
+Onolibcalls= オプション (新規) 12
+wendian オプション (新規) 12
+wlint オプション (拡張) 12
+wsecurity= オプション (拡張) 12
システム全体でのオプション設定 13
[NO]PTRS_TO_GLOBALS プラグマ 13
-AA -D_HP_NONSTD_FAST_IOSTREAM パフォーマンス改善マクロ 13
新しい関数属性 14
改良された診断機能 14
C++ 標準ライブラリの変更点 14
以前のバージョン 15

3. インストール情報

ハードウェア要件 18

4. 互換性に関する情報

aC++ の標準準拠と互換性に関する変更点 19
WDB の互換性 19
Caliper の互換性 20
32 ビットモードでコンパイルしたときと 64 ビットモードでコンパイルしたときのクラスサイズの違い 20
HP C++ (cfront) から HP aC++ への移行 21
一般的なプログラミング情報とサポートに関するお問い合わせ 21

目次

5. 既知の問題と回避策

最適化レベル +O4 または -ipo で作成されたオブジェクトファイル 23
C++ 標準ライブラリバージョン 1.2.1 とドラフト標準との非互換性. 24
macros.h と numeric_limits クラス (min および max) の競合 25
既知の制限事項 25

6. 関連マニュアル

オンラインマニュアル. 29
オンラインの C++ サンプルソースファイル 31
印刷ドキュメント. 31
その他のドキュメント 32
HP aC++ Web サイト 32
HP C Web サイト. 32

1 HP aC++/HP ANSI C リリースノート

このマニュアルに含まれる情報は、HP Integrity サーバー上の HP-UX 11i v2 および HP-UX 11i v3 オペレーティングシステム用の HP aC++ および HP ANSI C コンパイラのリリース、バージョン A.06.12 に関するものです。

HP ANSI C コンパイラは、ANSI programming language C standard ISO 9899:1990 をサポートしています。HP aC++ コンパイラは、ISO/IEC 14882 Standard for the C++ Programming Language (C++ の国際規格) をサポートしています。HP ANSI C および HP aC++ は、単一のソフトウェアバンドルとして配布されます。

このマニュアル内のすべての情報は、特に明記しない限り、HP C と HP aC++ の両方に適用されます。

このバージョンのコンパイラでは、報告された問題が修正されているほか、「新機能」の項に示す機能が追加または変更されています。

このマニュアルでは、以下のトピックについて説明します。

- 9 ページの「このバージョンの新機能」
- 17 ページの「インストール情報」
- 19 ページの「互換性に関する情報」
- 23 ページの「既知の問題と回避策」
- 29 ページの「関連マニュアル」

2 このバージョンの新機能

ここでは、このバージョンの HP aC++/HP C コンパイラで導入された新しいコマンド行オプションと機能の概要を説明します。

HP aC++ コンパイラのバージョン A.06.12 は、以前の A.06.xx バージョンファミリと完全なソースおよびバイナリ互換性があります (OpenMP 機能も含む)。HP C コンパイラのバージョン A.06.12 には、ごくわずかですが、以前のバージョンと互換性がない点があります。ただし、Integrity プラットフォームにアプリケーションを移行する場合は、これらの非互換性に伴う影響は、ほとんどありません。

バージョン A.06.12 の新機能

HP aC++ コンパイラのバージョン A.06.12 では、以下の機能が追加されています。

- オプション `+Ofast` (`-fast`) および `+Ofaster`
- `+Oinit_check` および `+check=uninit` 間の相互作用

オプション +Ofast (-fast) および +Ofaster

オプション `+Ofast` および `+Ofaster` を指定した場合は、パフォーマンスを改善する目的で、データおよびテキストページのサイズが 4 MB ではなく 1 MB に設定されるようになりました。また、`+Ofast` オプションは、安全でない可能性がある最適化 `+Ointeger_overflow=aggressive` および `+Olibcalls` を暗黙指定しなくなりました。

+Oinit_check および +check=uninit 間の相互作用

オプション `+Oinit_check` および `+check=uninit` を両方指定すると、コンパイル時には従来どおり、変数の初期化漏れの可能性を示す警告が出力されますが、コンパイラによる自動初期化は行われなくなりました。そのため、実行時に変数の初期化漏れが検出される可能性があります。

バージョン A.06.10 の新機能

HP aC++ コンパイラのバージョン A.06.10 は、以前の A.06.xx バージョンファミリと完全なソースおよびバイナリ互換性があります (OpenMP 機能も含む)。

HP aC++ コンパイラのバージョン A.06.10 では、以下の機能が追加、変更されています。

- HP Code Advisor
- +cond_rodata オプション (廃止)
- +[no]dep_name オプション (新規)
- +expand_types_in_diag オプション (新規)
- +FPmode (拡張)
- +Ointeger_overflow オプション (デフォルト値の変更)
- +Onolibcalls= オプション (新規)
- +wendian オプション (新規)
- +wlint オプション (拡張)
- +wsecurity= オプション (拡張)
- システム全体でのオプション設定 (新規)
- [NO]PTRS_TO_GLOBALS プラグマ (新規)
- -AA -D_HP_NONSTD_FAST_IOSTREAM パフォーマンスマクロ (新規)
- 新しい関数属性
- 改良された診断機能
- C++ 標準ライブラリ

HP Code Advisor

このリリースでは、C/C++ ソースコード内のさまざまなプログラミングエラーの検出に役立つ新しいツール "HP Code Advisor" が追加されています。このツールを使用すると、潜在的なコーディングエラー、移植時に問題となる箇所、セキュリティエラーを検出できます。

HP Code Advisor は、HP-UX PA-RISC サーバーおよび Integrity サーバーの両方で使用可能です。このツールは、Integrity サーバー用 C/C++ コンパイラの高度な解析機能を利用しています。

ツールの起動には、"/opt/cadvise/bin/cadvise" コマンドを使用します。次に示すように、-help オプション付きでコマンドを実行すると、簡単な説明が表示されます。

```
$ /opt/cadvise/bin/cadvise -help
```

追加情報は、http://www.hp.com/go/cadvise (英語) を参照してください。

+cond_rodata オプション (廃止)

このバージョンのコンパイラでは、より多くのデータを読み取り専用メモリに配置できます。そのため、このオプションは廃止されました。コンパイラは、このオプションが指定された場合の動作を常に実行するよう変更されています。(注記 : 以前のリリースノートでは、このオプションに関する説明文に誤植がありました。正しいオプション名は +cond_rodata です。)

+[no]dep_name オプション (新規)

+[no]dep_name オプションを指定すると、テンプレート内で厳格な依存名検索ルールが適用されます。デフォルトは +nodep_name です。

+expand_types_in_diag オプション (新規)

+expand_types_in_diag オプションは、診断内の typedef を拡大して、元の型と最終的な型の両方が存在できるようにします。

+FPmode オプション (拡張)

+FPmode は、浮動小数点処理のための実行時環境をプログラム開始時に初期化する方法を指定します。デフォルトでは、IEEE 浮動小数点標準で定義されているモードが使用されます。つまり、すべてのトラップは無効化され、漸近的アンダーフロー、最近似値への丸めが行われます。具体的なモード値の詳細は、ld(1) を参照してください。これらの設定を実行時に動的に変更する方法は、fenv(5)、fesettrapenable(3M)、fesetflushtozero(3M)、および fesetround(3M) を参照してください。

+Ointeger_overflow オプション (デフォルト値の変更)

すべての最適化設定に対するデフォルト値が "moderate" に変更されました。
+Ointeger_overflow=moderate

+Onolibcalls= オプション (新規)

+Onolibcalls= を使用すると、指定した関数について最適化をオフにできます。

+Onolibcalls=function1,function2,...

これにより、指定した関数呼び出しに対する libcall 最適化 (インライン化または置換) が無効化されます。この指定は、システムヘッダーファイルでの指定をオーバーライドします。

+wendian オプション (新規)

このオプションを使用すると、移植時にリトルエンディアン / ビッグエンディアン方式の違いによる問題が発生する可能性があるソース内領域を特定できます。

+wlint オプション (拡張)

このオプションに新しい診断機能が追加されました。これにより、メモリリーク、範囲外メモリアクセス、ヌルポインタ参照、境界外アクセスのチェックが可能になります。

+wsecurity= オプション (拡張)

+wsecurity オプションに、セキュリティメッセージの生成量を制御するための新しい引き数が追加されました。

+wsecurity=[1|2|3|4] (デフォルトレベルは 2)

システム全体でのオプション設定

システム全体でのコンパイラオプションの設定が可能になりました。以下の構成ファイルがコンパイラに読み込まれます (ファイルが存在する場合)。

/var/aCC/share/aCC.conf (aC++ 用)、または

/var/ansic/share/cc.conf (ANSI C 用)

次に示すように、構成ファイル内にオプションを指定する方法は、CCOPTS や CXXOPTS の場合と同じです。

[options-list-1] [| [options-list-2]]

options-list-1 に指定したオプションはコマンド行内のオプションよりも前に、
options-list-2 に指定したオプションはコマンド行内のオプションよりも後に適用されます。

最終的なオプションの適用順序は次のとおりです。

<file-options-1> <envvar-options-1> <command-line-options>

<envvar-options-2> <file-options-2>

"|" 文字より前の構成ファイルオプションはコンパイルのデフォルト設定を定義し、"|" 文字より後の構成ファイルオプションはユーザーのコマンド行設定をオーバーライドします。

注記 HP aC++ にはデフォルトの構成ファイルは付属していません。構成ファイルは、必要に応じてシステム管理者が設定してください。

[NO]PTRS_TO_GLOBALS プラグマ

#pragma [NO] PTRS_TO_GLOBALS <name>

エイリアス解析を支援するプラグマです。このプラグマはグローバルスコープで指定する必要があり、また、指定した変数またはエントリ宣言の直前に定義しなければなりません。プラグマからオプティマイザに、グローバル変数またはエントリの「名前」がポインター経由でアクセスされるかどうかが伝えられます。

-AA -D_HP_NONSTD_FAST_IOSTREAM パフォーマンス改善マクロ

新しいパフォーマンス改善プリプロセッサマクロ _HP_NONSTD_FAST_IOSTREAM は、-AA iostream のパフォーマンス改善に役立ちます。

このマクロにより、以下の非標準の機能が使用可能になります。

- std::ios_base::sync_with_stdio(false) を設定 (stdioとのデフォルトの同期を無効化)
- std::cin.tie(0) を設定 (cin を他のストリームから分離)
- "std::endl" をすべて "\n" に置換

iostreams を頻繁に使用するアプリケーションの場合は、このマクロを有効にすることで、パフォーマンスが大幅に向上する可能性があります。

注記 以下のいずれかに該当する場合は、このマクロを有効にしないでください。

- アプリケーションがC++ストリームとCストリームの同期を想定している場合
- アプリケーションが endl によるストリームフラッシュ動作に依存している場合
- ユーザーが出力ストリームの単位バッファリングに "std::cout.unsetf(ios::unitbuf)" を使用している場合

新しい関数属性

__attribute__ は、関数への属性の追加を可能にする言語拡張機能です。この拡張機能は #pragma による同種の機能と似ていますが、プラグマよりも言語構文への統合度が高く、ソースコード内の位置は属性の適用対象となる構造体によって異なります。

このリリースで追加された新しい関数属性は、"malloc"、"non-exposing"、"noreturn"、および "format" です。詳細は、オンラインヘルプのトピック「プラグマと属性」を参照してください。

改良された診断機能

診断メッセージの内容が強化され、改良されたテンプレートインスタンス生成トレースバックなど、根本原因を追跡するのに役立つ情報が追加されています。

C++ 標準ライブラリの変更点

技術正誤表 1 (Technical Corrigenda 1) により、STL 関数 make_pair は、const 参照ではなく値を引き数とするよう変更されています。この変更に伴い、HP ライブラリを標準に準拠させるためには、有効化マクロ -D__HP_TC1_MAKE_PAIR をコンパイル時に指定する必要があります。バイナリ互換性を維持するため、デフォルトの動作は変更されていません。

以前のバージョン

A.06.12 より以前のバージョンで導入された新機能については、各バージョンのリリースノートで確認してください。リリースノートは以下の HP テクニカルドキュメント Web サイトで入手できます。

http://docs.hp.com（英語版）

http://docs.hp.com/ja（日本語版）

3　インストール情報

インストールを開始する前に、このマニュアルおよびその他の関連リリースノートや readme ファイルをすべてお読みください。

ソフトウェアをインストールするには、SD-UX の `swinstall` コマンドを実行します。これにより、ユーザーインタフェースが起動するので、画面の指示に従ってインストールを行います。

インストール手順と、その関連事項についての詳細は、『*Software Distributor 管理者ガイド*』および HP-UX オペレーティングシステムのパッケージに付属のドキュメントを参照してください。

ご使用の環境によっては、ネットワーク、システムセキュリティ、ウィンドウシステムなど、付属していないマニュアルも必要な場合があります。

ハードウェア要件

重要

最適化レベル 2 (-O または +O2) 以上でファイルをコンパイルすると、コンパイラが必要とする仮想メモリーの量が大きくなります。+O2 で非常に大きな関数やファイルをコンパイルする場合や、積極的な最適化 (+O3 以上) を使用する場合には、コンパイルを実行するマシン上でカーネル調整パラメータ maxdsiz が適切に設定されていることを確認してください。

このような場合には、0x80000000 = 2GB に設定することをお勧めします (このパラメータのデフォルトは、0x40000000 = 1GB です)。maxdsiz を変更しておくことで、大きいファイルや関数のコンパイル時に、コンパイラが使用する仮想メモリーが不足する事態を防ぐことができます。さらに、非常に大きい、または複雑な入力ファイルの場合は、maxssiz を 128 MB に設定する必要があります。通常は、maxssiz のデフォルト値である 64 MB で十分です。

maxfiles の設定はデフォルト値の 1024 より小さくしないことをお勧めします。

カーネル調整パラメータの変更方法の詳細については、*kctune(1M)* マンページを参照してください。

HP aC++ は、/opt/aCC 内のファイル用に約 120MB を必要とします。

HP ANSI C は、/opt/ansic 内のファイル用に約 100 MB を必要とします。

その他の構成要素には、次のディスク容量が必要です。

- WDB 用に約 28 MB
- u2comp/be 用に約 60 MB
- HP Caliper 用に約 26 MB
- HP Code Advisor 用に約 105 MB

正確なサイズを確認するには、次のコマンドを実行します。

```
/usr/sbin/swlist -a size " 製品番号 "
```

4 互換性に関する情報

HP aC++ の新バージョンをリリースする際の主な要件は、バイナリ互換性の保持です。コンパイラは、同じオブジェクトモデルと呼び出し規約を保持します。また、HP aC++ およびそのランタイムパッチストリームのさまざまなリリースにおいて組み込み実行時ライブラリ (libCsup) と同時に生成されるコードにおいて HP-UX ランタイムとの互換性が保持されます。

aC++ の標準準拠と互換性に関する変更点

HP aC++ バージョン 5.x から HP aC++ バージョン 6.0 へのアップグレードに伴う影響の詳細は、次のドキュメントを参照してください。

http://h21007.www2.hp.com/dspp/tech/tech_TechSoftwareDetailPage_IDX/1,1703,7274,00.html (英語)

WDB の互換性

最新バージョンの WDB デバッガを使用することをお勧めします。デバッガの最新情報は、HP Web サイト http://www.hp.com/go/wdb の「Developer and Solution Partner」ページを参照してください。

Caliper の互換性

バージョン A.05.38 のコンパイラで作成されたオブジェクトを含むバイナリについては、パフォーマンスの計測と PBO に、Caliper バージョン 2.1 を使うことをお勧めします。Caliper 2.1 は、次の URL からダウンロードできます。

http://www.hp.com/go/caliper (英語)

32 ビットモードでコンパイルしたときと 64 ビットモードでコンパイルしたときのクラスサイズの違い

仮想関数を含むクラスのサイズは、32 ビットモードでコンパイルしたときと 64 ビットモードでコンパイルしたときで異なります。サイズの違いは、クラスオブジェクトの仮想テーブルポインター (内部コンパイラテーブルへのポインター) により生じます。(ポインターは 1 つ以上の仮想関数を含むクラスに対して作成されます。)

次の例を 32 ビットモードでコンパイルすると、8 が出力され、64 ビットモードでコンパイルすると、16 が出力されます。

```
extern "C" int printf(const char *,...);
   class A {
   int a;
public:
   virtual void foo();  //virtual function foo, part of class A
   };
void A::foo() {
   return;
   }
int main() {
   printf("%d\n",sizeof(A));
   }
```

HP C++ (cfront) から HP aC++ への移行

コンパイラは、エラー、予測されるエラー、警告を一覧表示します。HP aC++ へ移行することにより、プログラムコードに対して報告される警告、エラー、予測されるエラーの数は増えますが、多くは標準準拠の構文に関するエラーです。詳細は、『*HP aC++ Transition Guide*』を参照してください。このドキュメントは、以下の Web サイトから入手できます。

`http://www.hp.com/go/cpp`

一般的なプログラミング情報とサポートに関するお問い合わせ

一般的な背景知識や知見については、`cxx-dev` メーリングリストに登録してください。登録するには、本文に `subscribe list-name` と記したメッセージを `majordomo@cxx.cup.hp.com` あてに送信してください。

`cxx-dev` メーリングリストへの登録についての情報は、`http://www.hp.com/go/cpp` から入手できます。

指定可能なリスト名は次のとおりです。

`cxx-dev`	『*HP C++ Development Discussion List*』
`cxx-dev-announce`	『*HP C++ Development Announcements*』
`cxx-dev-digest`	『*HP C++ Development Announcements*』

さらに詳しい内容やメーリングリストに関する情報が必要な場合は、本文に `help` と記したメッセージを `majordomo@cxx.cup.hp.com` あてに送信してください。

特定のサポートに関するご質問は、当社サポート担当者にお問合せください。

一般的な C++ に関するご質問は、『*HP aC++ オンライン プログラマーズガイド*』の情報マップに記載されているドキュメントおよび URL を参照してください。

5 既知の問題と回避策

この項では、既知の問題と回避策について説明します。サポート契約を締結しているユーザーは、製品番号を使って HP aC++ または HP C 用の SSB および SRB レポートを見つけることができます。検索する製品番号は B3910BA です。

ご使用の HP aC++ または HP C の製品番号とバージョンを確かめるには、次のような HP-UX コマンドを実行します。

```
what /opt/aCC/bin/aCC

what /opt/aCC/lbin/ecom

what /opt/ansic/bin/cc

what /opt/ansic/lbin/ecom
```

最適化レベル +O4 または -ipo で作成されたオブジェクトファイル

コンパイラの最適化レベル 4 で作成されたオブジェクト (中間オブジェクトファイル) は、一時ファイルとしてのみ使えます。これらのオブジェクトファイルには、ユーザーコードの中間表現が、高度な最適化のために設計されたフォーマットで入っています。これらの中間オブジェクトのファイルサイズは、通常のオブジェクトファイルの 3 ～ 10 倍程度の大きさになります。当社は、コンパイラの新版のリリースやパッチに伴ってこれらのファイルのフォーマットを予告なしに変更する権利を、留保しています。中間ファイルの使用は、そのファイルを作成したコンパイラだけに限らなければなりません。同じ理由で、異なるバージョンのコンパイラで使われる可能性のあるアーカイブライブラリには、中間オブジェクトファイルを入れないでください。

互換性のない中間ファイルを検出した場合、コンパイラはメッセージを出力して終了します。

詳細は、「インストール情報」の項の調整パラメータに関する説明を参照してください。

C++ 標準ライブラリバージョン 1.2.1 とドラフト標準との非互換性

ANSI C++ 標準は度々改訂されていますが、改訂の都度、C++ 標準ライブラリが変更されている訳ではありません。関数ヘッダーファイル functional の `times` 関数オブジェクトなどがその例です。ANSI C++ 標準では、`times` は `multiplies` に名前が変更されています。

`multiplies` を使用してコードを記述する場合は、ISO/ANSI C++ 標準と互換性を持つように、aCC コマンド行で条件付きコンパイルフラグを使用してください。

たとえば、次のプログラムであれば、次のようなコマンド行でコンパイルしてください。

```
    aCC -D__HPACC_USING_MULTIPLIES_IN_FUNCTIONAL test.c

// test.c

int times;                    //user defined variable

#include <functional>

    // multiplies can be used in

int main() {}

// end of test.c
```

現在の A.05.* および A.06.* コンパイラは自動的に次のようなフラグを設定します。

- `-D__HPACC_USING_MULTIPLIES_IN_FUNCTIONAL`
- `-D__HPACC_THREAD_SAFE_RB_TREE`
- `-D__HPACC_USING_MULTIPLIES_IN_FUNCTIONAL`
- `-D__HPACC-FIX_FUNC_ADAPTER_OPERATOR`
- `-D__HPACC_FULL_ITERATOR_REL_OPS`
- `-D__HPACC_TEMPLATE_PAIR_CTOR`
- `-D__HPACC_MEM_FUN_ADAPTOR`

macros.h と numeric_limits クラス (min および max) の競合

コードに /usr/include/macros.h が含まれる場合は、macros.h で定義されている min マクロおよび max マクロが、C++ 標準ライブラリの numeric_limits クラスで定義されている min 関数および max 関数と競合することに注意してください。たとえば、次のコードはコンパイラ Error 134 を生成します。

```
numeric_limits<unsigned int>::max();
```

macros.h ヘッダーを使わなければならない場合は、競合するマクロを定義しないようにしてください。

```
...
#include <macros.h>
#undef max
#undef min
...
```

既知の制限事項

このリリースに関する既知の制限事項を以下に示します。これらの制限事項のいくつかは、HP aC++ の今後のリリースで取り除かれる予定です。制限事項の中には特定のプラットフォームでのみ生じるものがある点に注意してください。

- HP aC++ の <iostream.h> および <iostream> は、大容量 (2 GB を超える) ファイルをサポートしていません。
- 直接参照されていないオブジェクトについては、シンボリックデバッグ情報が必ず出力されるとは限りません。たとえば、オブジェクトへのポインターが使用されているにもかかわらずフィールドが参照されていない場合、HP aC++ は、ポインターの型のシンボリックデバッグ情報のみを出力し、ポインターが示すオブジェクトの型については出力しません。たとえば、Widget * を使用すると、ポインター型 Widget * のデバッグ情報が出力されるだけで、Widget は出力されません。そのような情報が必要な場合は、その型 (Widget) のパラメータを持つダミー関数を定義するソースファイルを新たに作成し、実行可能プログラムにリンクします。

- 例外処理機能の既知の制限事項
  - setjmp/longjmp との相互運用性 (C++ 国際標準試案でも未定義) は実装されていません。longjmp を実行しても、デストラクタは実行されません。
  - プログラムの初期化段階 (つまり、メインプログラムの実行前) に未処理の例外が生じた場合、生成されたオブジェクトによってはデストラクタが実行されない場合があります。
  - HP aC++ は、リンカーオプション -Bsymbolic をサポートしていません。共有ライブラリから型を削除する場合は、コンパイル時オプション -Bprotected_def を使用してください。この制限事項は、dynamic_cast および RTTI にも適用されます。
- シグナル処理機能の既知の制限事項
  - シグナルハンドラーからの例外の送出はサポートされていません。これは、シグナルは、破壊可能オブジェクトの値がレジスタ内に一時的に保存されているようなコードの最適化された領域など、どこででも発生する可能性があるためです。例外処理はメモリー内で更新中の破壊可能オブジェクトによって決まりますが、この状態が保障されているのは、呼び出し場所だけです。
  - 例外の発生がサポートされていないため、シグナルハンドラーで longjmp を実行することも同様にお勧めしません。シグナルハンドラーがコードの処理を中断すると、データ構造が未定義となって、予測不可能な結果に陥ります。
- C++ 共有ライブラリのソースレベルのデバッグはサポートされています。ただし、メンバー関数が共有ライブラリで宣言されているクラスで、そのクラスが定義されている共有ライブラリ以外で宣言されているオブジェクトを持つものである場合、C++ 共有ライブラリのデバッグに関して制限があります。オペレーティングシステムおよびデバッガのリリースノートおよびマニュアルを参照してください。追加の詳細情報については、Software Status Bulletin も参照してください。
- 共有メモリー上の仮想関数での共有オブジェクトのインスタンス生成はサポートしていません。
- ライブラリレベルのバージョニングでの shl_load(3X) または dlopen(3C) の使用

  ライブラリレベルのバージョニングを行うと、shl_load() または dlopen()(shl_load(3X) を参照) への呼び出しでは、ロードするライブラリの実際のバージョンを指定する必要があります。

  たとえば、libA.so が現在 libA.so.1 のシンボリックリンクの場合、このライブラリをダイナミックにロードする呼び出しでは、アプリケーションがコンパイルされたときの最新バージョンを指定する必要があります。たとえば、次のようになります。

```
shl_load("libA.so.1", BIND_DEFERRED, 0);
```

つまり、libA よりも後のバージョンを利用できるシステムにアプリケーションが移行された場合、実際に必要なバージョンはダイナミックにロードされるものであることが保証されます。

- メモリー割り当てルーチン alloca()

  コンパイラは、/usr/include/alloca.h ヘッダーファイルで定義されているビルトイン関数 alloca をサポートしています。alloca() ルーチンの実装はシステムに依存するので、使用はお勧めしません。

  alloca() は malloc() と同様のメモリー割り当てルーチンです (malloc(3C) を参照)。構文は次のとおりです。

  ```
  void *alloca(size_t <size>);
  ```

  alloca() は呼び出し元のスタックから、少なくとも <size> バイトのブロックに領域を割り当てますが、領域の初期化は行いません。呼び出しルーチンが終了すると、領域は自動的に解放されます。

  注記 : alloca() で返されたメモリーは、他のメモリー割り当て関数によって割り当てられたメモリーとは関連がありません。alloca() で返されたアドレスを、他のメモリー関数のパラメータとして渡したときの動作は定義されていません。

  この関数を使用するには、<alloca.h> ヘッダーファイルを使用しなければなりません。

6 関連マニュアル

この章では、HP aC++ / HP C に関連するマニュアルについて説明します。

オンラインマニュアル

HP aC++ / HP C 製品には、次に示すオンラインマニュアルが用意されています。

- 『*HP aC++ オンライン プログラマーズガイド*』

  次のいずれかの方法を使用して、マニュアルを参照してください。

  — 次のように、+help コマンド行オプションを使います。

  ```
  /opt/aCC/bin/aCC +help
  ```

  — Web ブラウザで次に示す URL を参照します。

  — `file:/opt/aCC/html/C/guide/index.htm` (英語版)

  — `file:/opt/aCC/html/ja_JP.SJIS/guide/index.htm` (日本語版)

  — このガイド (Rogue Wave のマニュアルを除く) は次の URL でも参照できます。

  — `http://docs.hp.com/hpux/dev` (英語版)

  — `http://docs.hp.com/ja/hpuxdev` (日本語版)

- 『*HP-UX 64-bit Porting and Transition Guide*』

  このマニュアルは、開発者がアプリケーションを HP-UX 32 ビットプラットフォームから HP-UX 64 ビットプラットフォームへ移行する際に役立ちます。このマニュアルは HP-UX 11i/11.0 CD-ROM および次の URL で参照できます。

  `http://www.hp.com/go/cpp`

- 『*HP-UX Linker and Libraries User's Guide*』

  次のコマンドを使い、マニュアルを参照します。

  ```
  /usr/ccs/bin/ld +help
  ```

  日本語版マニュアルは以下のサイトで入手できます。

http://docs.hp.com/ja/hpuxdev

- 『*HP Wildebeest Debugger (HP WDB)*』

  HP WDB ドキュメントは、/opt/langtools/wdb/doc にあります。

  HP WDB および関連マニュアルは、以下の Web サイトから入手できます。

  http://www.hp.com/go/wdb

- 『*Rogue Wave Software Standard C++ Library 2.2.1 Class Reference*』

  このリファレンスには、更新された Rogue Wave Standard C++ Library のクラス、アルゴリズム、および関数オブジェクトがアルファベット順に記載されています。ライブラリには、標準の iostream ライブラリが含まれ、また**名前空間** std が使用可能です。

  リファレンスは HTML 形式のファイルです。これらのファイルを参照するには、HTML ブラウザでファイル /opt/aCC/html/libstd_v2/stdref/index.htm を開くか、『*HP aC++ オンライン プログラマーズガイド*』からのハイパーリンクを選択します。

- 『*Rogue Wave Software Standard C++ Library 2.2.1 User's Guide*』

  このガイドにはライブラリの使用法に関する情報、およびロケールと iostreams に関する詳細な説明が記載されています。

  このガイドは HTML 形式のファイルです。これらのファイルを参照するには、HTML ブラウザでファイル /opt/aCC/html/libstd_v2/stdug/index.htm を開くか、メインオンラインヘルプのトピック内で「*HP aC++*」へのハイパーリンクを選択します。

- 『*Rogue Wave Software Standard C++ Library 1.2.1 Class Reference*』

  このリファレンスには、以前の Rogue Wave 版標準 C++ ライブラリのすべてのクラス、アルゴリズム、および関数オブジェクトがアルファベット順に記載されています。リファレンスは HTML 形式のファイルです。これらのファイルを参照するには、HTML ブラウザで /opt/aCC/html/libstd/ref.htm を開きます。

- 『*Rogue Wave Software Tools.h++ 7.0.6 Class Reference*』

  このリファレンスには、Tools.h++ Library のすべてのクラスおよび関数がアルファベット順に記載されています。このリファレンスは Rogue Wave Standard C++ Library 1.2.1 を対象としています。

  リファレンスは HTML 形式のファイルです。これらのファイルを参照するには、HTML ブラウザで /opt/aCC/html/librwtool/ref.htm を開きます。

マニュアルの主な部分に、「サポート中」として 8 つのテンプレートが記載されていますが、これは正しい情報ではありません。次の 8 つのテンプレートへのインタフェースは、2 つの追加のテンプレート引き数で新しいテンプレートに変換しなければなりません。

```
RWTPtrHashDictionary          ==> RWTPtrHashMap
RWTPtrHashDictionaryIterator ==> RWTPtrHashMapIterator
RWTPtrHashTable               ==> RWTPtrHashMultiSet
RWTPtrHashTableIterator       ==> RWTPtrHashMultiSetIterator
RWTValHashDictionary          ==> RWTValHashMap
RWTValHashDictionaryIterator ==> RWTValHashMapIterator
RWTValHashTable               ==> RWTValHashMultiSet
RWTValHashTableIterator       ==> RWTValHashMultiSetIterator
```

不具合番号 JAGaa90638 を参照してください。

注記 Rogue Wave に関する追加ドキュメントの入手方法については、『*HP aC++ オンライン プログラマーズガイド*』の情報マップを参照してください。

- 『*HP aC++ リリースノート*』はこのマニュアルです。このマニュアル (ただし、英語版) のオンライン テキストファイルが格納されている場所は次のとおりです。

  `/opt/aCC/newconfig/RelNotes/ACXX.release.notes`

- aCC および c++filt 用オンラインマンページの場所は `/opt/aCC/share/man/man1.Z` (英語版)、`/opt/aCC/share/man/ja_JP.*/man1.Z` (日本語版) です。

  C++ 標準ライブラリおよび cfront 互換性ライブラリ (iostream および標準コンポーネント) のマンページの場所は `/opt/aCC/share/man/man3.Z` (英語版)、`/opt/aCC/share/man/ja_JP.*/man3.Z` (日本語版) です。

オンラインの C++ サンプルソースファイル

オンラインの C++ サンプルソースファイルは、`/opt/aCC/contrib/Examples/RogueWave` ディレクトリにあります。標準 C++ ライブラリおよび Tools.h++ ライブラリ用のサンプルも含まれます。

印刷ドキュメント

このマニュアルの英語版『HP aC++/HP ANSI C Release Notes』が、HP aC++ / HP C 製品に同梱されています。上記で説明したとおり、このマニュアルはオンライン形式でも利用可能です。

その他のドキュメント

C++ 言語に関するドキュメント一覧、URL、コース情報については、『*HP aC++ オンライン プログラマーズガイド*』の情報マップを参照してください。

HP aC++ の参考として、次のようなドキュメントも用意されています。

- 『*HP-UX パラレル プログラミング ガイド*』に、HP-UX 上で HP Fortran 90、HP C、および HP aC++ コンパイラ用に使用できる効果的なパラレルプログラミング手法が記載されています。

  このマニュアルは HP-UX 11i Instant Information メディアおよび次の URL で参照できます。

  `http://docs.hp.com/hpux/dev` (英語版)

  `http://docs.hp.com/ja/hpuxdev` (日本語版)

HP aC++ Web サイト

HP aC++ の Web サイトは、次の URL で参照できます。

`http://www.hp.com/go/cpp`

HP aC++ Web サイトには、次のような最新情報が記載されています。

- よくある質問 (FAQ)
- リリースバージョンおよびパッチの一覧
- サポート情報
- ドキュメントのリンク (リリースノートおよびプログラミングガイドも含む)
- リリース間の互換性。

HP C Web サイト

HP C の Web サイトは、次の URL で参照できます。

- `http://www.hp.com/go/c`